Financial Data Analysis

Overview of the Course

Nicholas Hirschey

Nova SBE

Introduction

- Welcome!

- Who am I?
  - Name: Nicholas Hirschey
  - Joined Nova this fall. Previously on the faculty at LBS 2012-2020.
  - PhD in Finance from the University of Texas at Austin.
  - Research interests
    - Asset Pricing (what should a stocks expected return be?)
    - Investments (trading by hedge funds/mutual funds/individuals)
    - Microstructure (HFT; where does information come from and how does it get into prices?)

Agenda for today

- Class Overview
- Requirements and Expectations
- First topic: Fundamentals of programming with financial data.

Course

- This course will teach you modern, high-performance computer programming for financial data analysis.
  - You will learn to write your own custom computer code to analyze real stock and bond market data.
  - We will cover in depth forming portfolios and analyzing their performance.
  - The type of analysis performed by researchers trying to find strategies at long-short quantitative hedge funds.
    - AQR, Renaissance Institutional Equities Fund (their fund for outsiders), etc.
- The course will go deep, but it is deliberately designed to be accessible.
  - I don't expect any prior programming experience. We start at the beginning and go from there.
  - The first lecture especially emphasizes programming basics, but this course is mostly about ***financial data analysis***.
  - If you have prior programming experience, be patient the first few lectures.

Course outline (subject to change)

1. Programming basics, returns, and volatilities.
2. Volatility modelling and volatility timing strategies.
3. Stock market return predictability.
4. Constructing mean-variance efficient portfolios.
5. Factor models and factor hedging.
6. Estimating factor loadings.
7. Portfolio performance evaluation.
8. Advanced portfolio optimization.
9. Constructing trading strategies from a signal.
10. Testing trading strategies.
11. Constructing "Smart Beta" Portfolios.
12. Review

Your grade is determined by

- Mid-term/Quizzes (35%)

- Final Project (65%)
  - Construct and analyze an investment strategy.

Contacting me

- Email: **nicholas.hirschey@novasbe.pt**
- Office Hours—by appointment
- The course's Moodle discussion forum is the best place to ask questions.
  - It's likely other people have your same question.
  - If the questions/answer are in the forum, other students can benefit.
  - It's possible another student will be able to answer your question faster than me or the TAs.
- Nova Moodle:
  - Lecture notes and assignments
  - Background readings and articles

The TAs

- Felix Brunner felix.brunner@novasbe.pt
- Robert Hill robert.hill@novasbe.pt

- Asking questions through the Moodle discussion forum is preferred.

What can you expect from me?

1. Lecture notes.
   - Some notes/code will contain some extra information that is not covered in class (but for you to read).
   - This is for you to be “ahead of the curve”.
   - Some occasional readings for you on moodle.
2. Questions and concerns will be promptly addressed.
   – We have excellent TAs.
3. Quizzes are graded promptly.

What I expect of you?

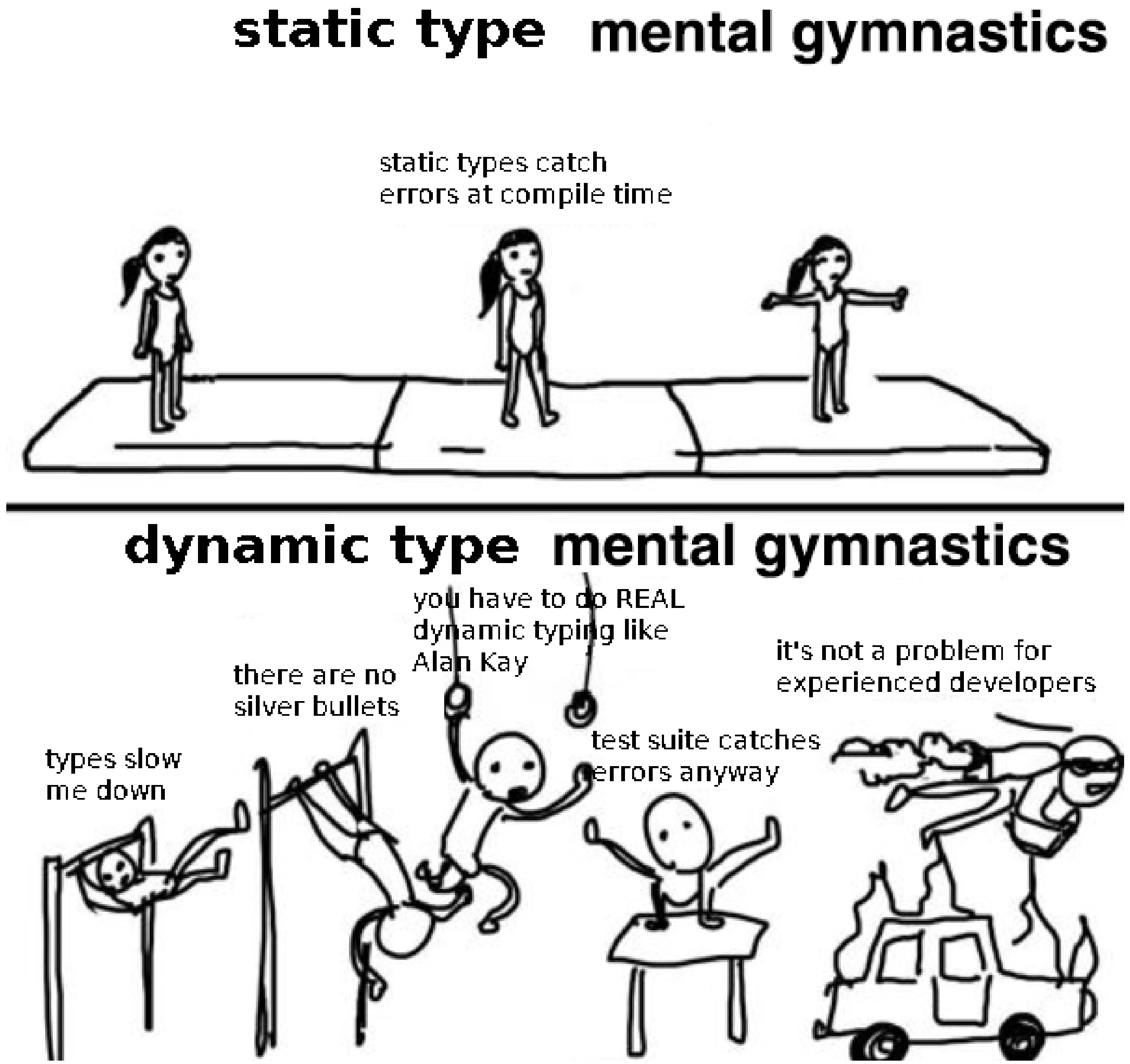
- Come to class prepared and on time.
- Participate in class
  - I will cold call sometimes: make sure you stay awake.
  - Ask questions!
  - Please come to class on time.
  - Have video on when you can, it helps me get to know you!
- Absolutely no late assignments will be accepted.
- Please follow the Nova “honor code”.
- Have fun!!

We will emphasize F#

- “An open-source, cross-platform, strongly-typed, succinct programming language with broad applicability to many different programming scenarios.” – Don Syme, creater
- Built on the high-performance .NET platform with broad library support.
- You get the speed of C++/Java with the simplicity of Python.
  - See Google’s Tensorflow motivation for moving away from Python to Swift.
- An advanced language with many uses
  - Data analysis scripts/notebooks
  - High-performance trading systems
  - iOS and Android mobile apps
- Industrial users: Credit Suisse, Walmart labs, Kaggle, …

Statically-typed functional programming: the sweet spot

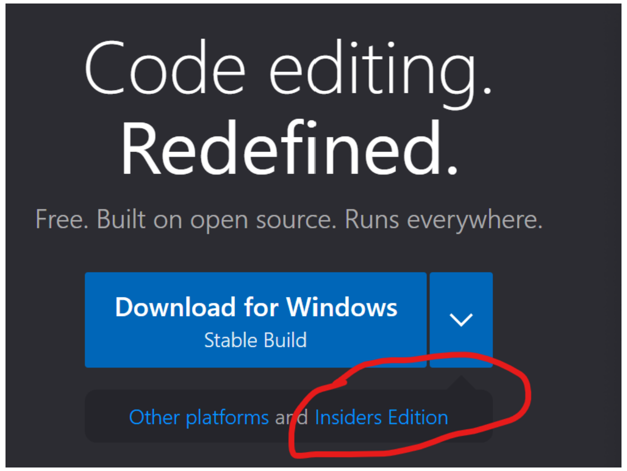
- The functional programming paradigm is growing in popularity
  - Swift (Apple), ReasonML (Facebook), F# (Microsoft), OCaml (Janestreet Trading)
  - Especially useful in distributed/multi-core processing (e.g., MapReduce).
- Static typing helps you write correct code. Other languages are now trying to add it after the fact.
  - "At our scale—millions of lines of Python—the dynamic typing in Python made code needlessly hard to understand and started to seriously impact productivity. To mitigate this, we have been gradually migrating our code to static type checking using mypy." – Dropbox (Our journey to type checking 4 million lines of Python – Dropbox)
  - "…in October 2017 a small team of engineers conceived of building Sorbet, a gradual static type system for Ruby. Static type systems look for certain classes of potential errors without running your code" – Stripe (State of Sorbet Spring 2019 · Sorbet)
- F# is built around a static type system from the start. This facilitates advanced tooling and performance.


static type mental gymnastics
static types catch errors at compile time
dynamic type mental gymnastics
types slow me down
there are no silver bullets
you have to do REAL dynamic typing like Alan Kay
test suite catches errors anyway
it's not a problem for experienced developers

- Enough talk.
- Let’s code.

We need to install some tools

- Install F# tools.
  - The .NET Core SDK. This includes the F# compiler and F# interactive.
  - Visual Studio Code. This is a good multi-purpose code editor (MSWord for code).
  - Ionide. An excellent integrated development environment for F#.
  - How to install?
  - Choose “Option 2: Install Visual Studio Code and Ionide” below.
    - Choose the “Insider Edition” of Code (it has some features we’ll use later).
    - https://fsharp.org/use/windows/
    - https://fsharp.org/use/mac/


Code editing.
Redefined.
Free. Built on open source. Runs everywhere.
Download for Windows
Stable Build
Other platforms and Insiders Edition

- Github
  - Sign up for a github account: www.github.com
  - Install Github desktop: https://desktop.github.com/